Amendments to the Claims:

1. (currently amended) A method for coordinating execution of compound Web services over a network, the method comprising the computer-implemented steps of:
receiving a request for [[the]] a particular compound Web service that uses output from at least a first Web service and a second Web service;
reading dependency information [[associated with]] specific to the particular compound Web service, wherein the dependency information specifies a sequence in which a plurality of modules must be executed to perform the particular compound Web service;
wherein properties of the plurality of modules are defined and set specifically for the particular compound Web service at service design time and persist through service runtime;
based on the dependency information, coordinating execution of the plurality of modules in the specified sequence, wherein the modules [[including]] include at least a first service execution module having a module property that is set specifically for coordinating execution of the first Web service and a second service execution module having a module property that is set specifically for coordinating execution of the second Web service;
controlling execution of the particular compound Web service using the properties of the plurality of modules defined and set for the particular compound Web service; and
generating a result of the compound Web service based on the output from the first and second Web services.

2. (original) The method of claim 1 wherein the dependency information specifies the sequence as a directed graph and the step of coordinating execution of the plurality of modules in the sequence includes coordinating according to information represented by the directed graph.

3. (original) The method of claim 1 wherein the step of coordinating execution of the first service includes coordinating execution of a second compound service that uses output from at least a third service and a fourth service.

4. (original) The method of claim 1 wherein the request for execution of the compound service is received from a source, the method further comprising the step of: transmitting the result of the compound service to the source.

5. (original) The method of claim 1 wherein the request for execution of the compound service is received from a source, the method further comprising the step of: transmitting the result of the compound service to a destination different than the source.

6. (original) The method of claim 1 wherein the dependency information specifies a sequence in which a transformation module must be executed to perform the compound service and the step of coordinating execution of the plurality of modules in the sequence includes coordinating execution of the transformation module to transform information received by the transformation module from a first data structure to a second data structure.

7. (original) The method of claim 1 wherein the dependency information specifies a sequence in which a splitter module must be executed to perform the compound service and the step of coordinating execution of the plurality of modules in the sequence includes coordinating execution of the splitter module to divide a message received by the splitter module into a plurality of messages.

8. (original) The method of claim 1 wherein the dependency information specifies a sequence in which a merger module must be executed to perform the compound service and the step of coordinating execution of the plurality of modules in the sequence includes coordinating execution of the merger module to merge a plurality of messages received by the merger module into a single message.

9. (original) The method of claim 1 wherein the dependency information specifies a sequence in which a conditional module must be executed to perform the compound service and the step of coordinating execution of the plurality of modules in the sequence includes coordinating execution of the conditional module to determine which module to execute next based on a condition status.

10. (original) The method of claim 1 wherein the dependency information specifies a sequence in which one or more of the plurality of modules must be executed concurrently to perform the compound service and the step of coordinating execution of the plurality of modules in the sequence includes coordinating concurrent execution of the one or more modules.

11. (currently amended) A computer-readable medium [[carrying]] storing one or more sequences of instructions for coordinating execution of compound service over a network, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

receiving a request for [[the]] a particular compound Web service that uses output from at least a first Web service and a second Web service;

reading dependency information [[associated with]] specific to the particular compound Web service, wherein the dependency information specifies a sequence in which a plurality of modules must be executed to perform the particular compound Web service;

wherein properties of the plurality of modules are defined and set specifically for the particular compound Web service at service design time and persist through service runtime;

based on the dependency information, coordinating execution of the plurality of modules in the specified sequence, wherein the modules [[including]] include at least a first service execution module having a module property that is set specifically for coordinating execution of the first Web service and a second service execution module having a module property that is set specifically for coordinating execution of the second Web service;

controlling execution of the particular compound Web service using the properties of the plurality of modules defined and set for the particular compound Web service; and

generating a result of the compound Web service based on the output from the first and second Web services.

12. (original) The computer-readable medium of claim 11 wherein the dependency information specifies the sequence as a directed graph and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating according to information represented by the directed graph.

13. (original) The computer-readable medium of claim 11 wherein the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating execution of a second compound service that uses output from at least a third service and a fourth service.

14. (original) The computer-readable medium of claim 11 wherein the request for execution of the compound service is received from a source and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of:

transmitting the result of the compound service to the source.

15. (original) The computer-readable medium of claim 11 wherein the request for execution of the compound service is received from a source and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of:

transmitting the result of the compound service to a destination different than the source.

16. (original) The computer-readable medium of claim 11 wherein the dependency information specifies a sequence in which a transformation module must be executed to perform the compound service and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating execution of the transformation module to transform information received by the transformation module from a first data structure to a second data structure.

17. (original) The computer-readable medium of claim 11 wherein the dependency information specifies a sequence in which a splitter module must be executed to perform the compound service and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating execution of the splitter module to divide a message received by the splitter module into a plurality of messages.

18. (original) The computer-readable medium of claim 11 wherein the dependency information specifies a sequence in which a merger module must be executed to perform the compound service and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating execution of the merger module to merge a plurality of messages received by the merger module into a single message.

19. (original) The computer-readable medium of claim 11 wherein the dependency information specifies a sequence in which a conditional module must be executed to perform the compound service and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform

the step of coordinating execution of the plurality of modules in the sequence including coordinating execution of the conditional module to determine which module to execute next based on a condition status.

20. (original) The computer-readable medium of claim 11 wherein the dependency information specifies a sequence in which one or more of the plurality of modules must be executed concurrently to perform the compound service and the execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the step of coordinating execution of the plurality of modules in the sequence including coordinating concurrent execution of the one or more modules.

21. (currently amended) A [[system]] computer program product [[comprising]] storing one or more sequences of instructions executable by a computer, comprising:

a plurality of invokable modules, wherein at least a subset of the plurality of invokable modules are configurable by setting one or more properties for controlling execution of corresponding particular ~~are associated with~~ compound Web services;

wherein properties of the plurality of invokable modules are defined and set for corresponding particular compound Web services at compound Web service design time and persist through compound Web service runtime; and

a compound service execution adapter configured to coordinate execution of the invokable modules according to particular dependency information corresponding to particular compound Web services, wherein the dependency information [[that]] specifies a sequence in which the plurality of invokable modules, with at least one of the one or more properties set, must be executed to perform [[a]] particular compound Web services ~~in response to a received request for a compound service~~ by utilizing properties defined and set specifically for the particular compound Web services.

22. (original) The system of claim 21 wherein the dependency information is a directed graph.

23. (original) The system of claim 21 wherein the request received by the compound service execution adapter is converted to an Extensible Markup Language (XML) file by the compound service execution adapter.

24. (original) The system of claim 21 wherein the plurality of invokable modules includes one or more message transformation modules configured to transform a first arrangement of received message information into a second arrangement of message information.

25. (original) The system of claim 24 wherein the received message information is an Extensible Markup Language (XML) message and is transformed by applying an Extensible Stylesheet Language Transformation (XSLT) stylesheet to the XML message.

26. (original) The system of claim 21 wherein the plurality of invokable modules includes one or more message splitter modules configured to divide an input message into a plurality of output messages.

27. (original) The system of claim 26 wherein the input message is an Extensible Markup Language (XML) message and is divided by applying an Extensible Stylesheet Language Transformation (XSLT) stylesheet to the XML message.

28. (original) The system of claim 21 wherein the plurality of invokable modules includes one or more message merger modules configured to merge a plurality of input messages into a single output message.

29. (original) The system of claim 28 wherein the input messages are Extensible Markup Language (XML) messages and are merged by applying an Extensible Stylesheet Language Transformation (XSLT) stylesheet to the XML message.

30. (original) The system of claim 21 wherein the plurality of invokable modules includes one or more service execution modules configured to interpret one or more received

messages as a request for a second compound service constituent to the compound service.

31. (original) The system of claim 21 wherein the plurality of invokable modules includes one or more conditional modules configured to direct the compound service execution adapter according to a condition status as to which invokable module to next execute.

32. (original) The system of claim 21 wherein the dependency information specifies a sequence in which one or more of the plurality of invokable modules must be executed concurrently to perform the compound service and the compound service execution adapter coordinates concurrent execution of the one or more invokable modules according to the sequence.

33. (currently amended) A method for managing execution of a compound Web service that is defined using the Web Service Definition Language (WSDL), the method comprising the computer-implemented steps of:

[[defining the]] specifying a particular compound Web service by generating a directed graph that indicates a sequence in which a plurality of base Web services must be executed to perform the particular compound Web service, wherein each node of the graph corresponds to a module, and wherein each module includes one or more properties and communicates with one or more other modules via one or more event messages, and wherein the one or more properties ~~contribute to~~ are set at compound Web service design time specifically for controlling execution of the plurality of base services and persist through compound Web service runtime; and

wherein at least one module is from a set consisting of:

a service execution module that, based on the module properties set specifically for the particular compound Web service, interprets one or more input messages as a request for the compound Web service and generates a result output message based on one or more responses from the plurality of base Web services, a message transformation module that transforms, based on an associated module property that specifies a particular one or more XSLT stylesheets specifically for transforming data for the particular compound Web service, an input data structure of one or more event messages into an output data structure different than the input data structure,

a message splitter module that divides, based on an associated module property that specifies a particular one or more XSLT stylesheets specifically for splitting input messages for the particular compound Web service, an input message into a plurality of output messages, and

a message merger module that merges, based on an associated module property that specifies a particular one or more XSLT stylesheets specifically for merging output messages associated with the particular compound Web service, a plurality of input messages into a single output message.

34. (new) The method of Claim 1, wherein the compound, first and second Web services are defined using Web Service Definition Language (WSDL).

35. (new) The method of Claim 1, wherein at least one of the first and second Web services are developed by a first party and wherein the compound Web service is developed by a second party that is different from the first party.